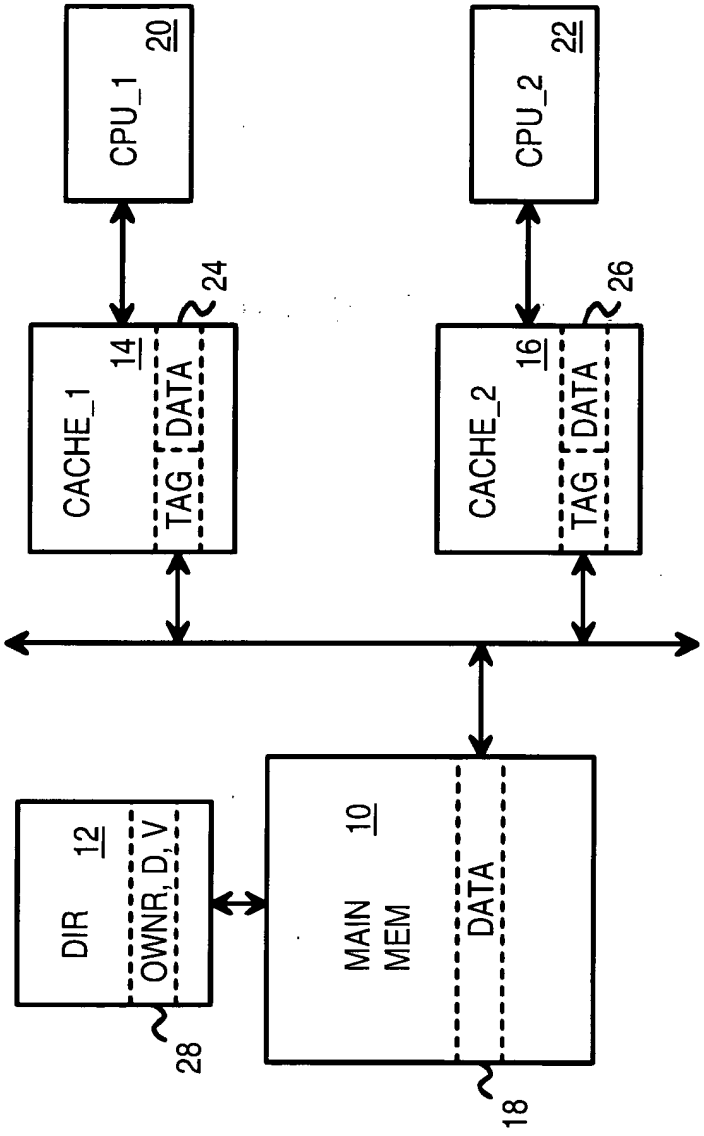


FIG. 1

PRIOR ART

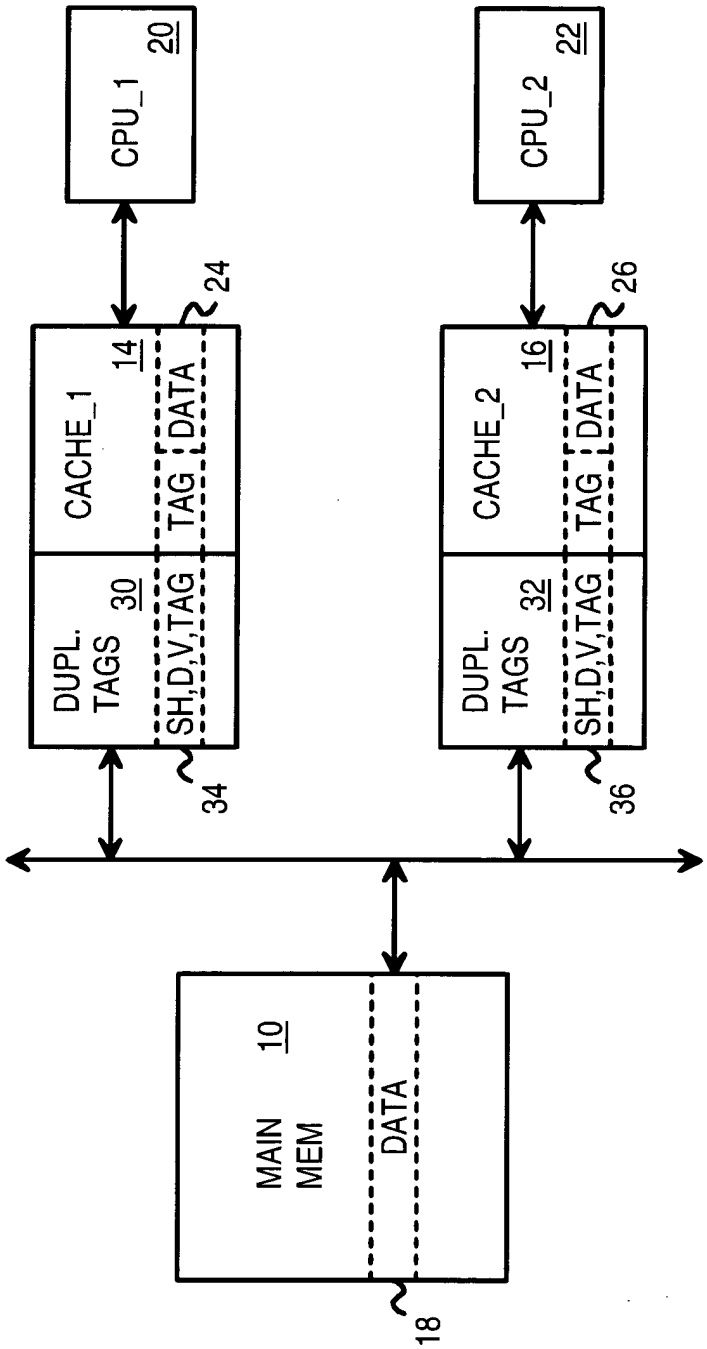


FIG. 2

PRIOR ART

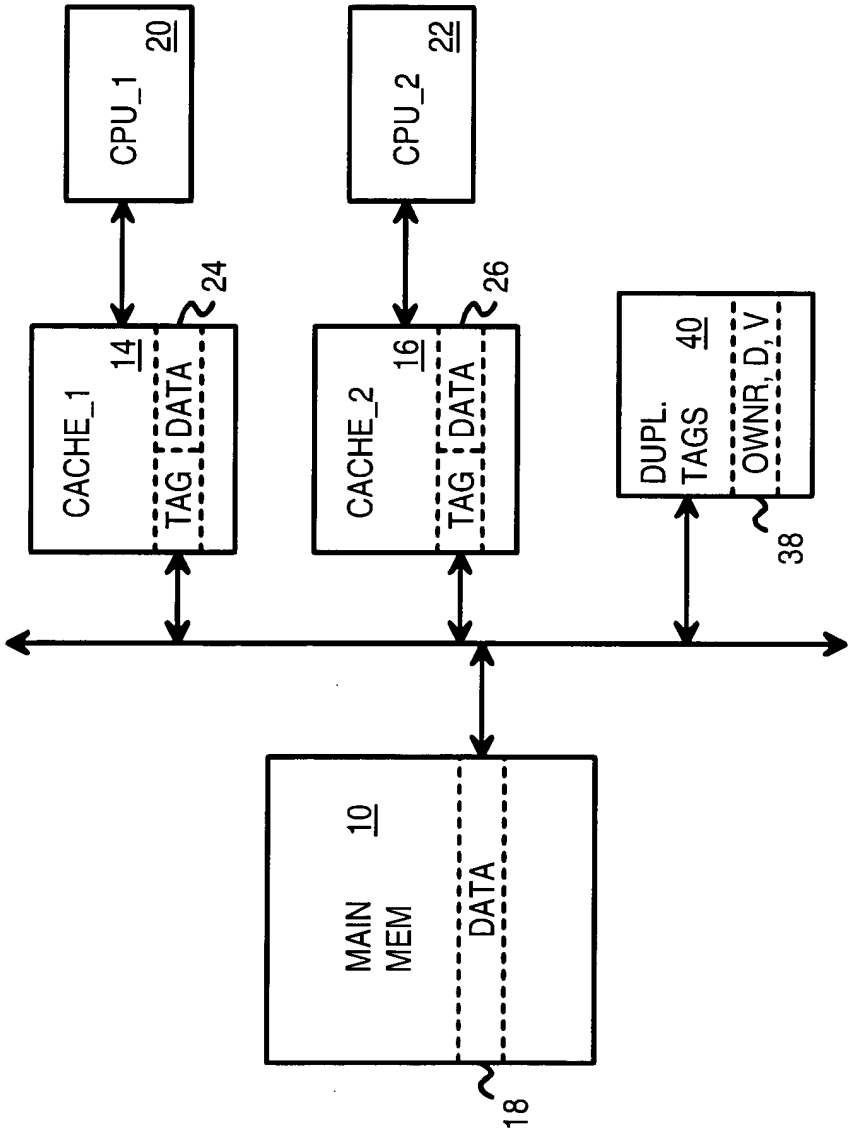


FIG. 3

PRIOR ART

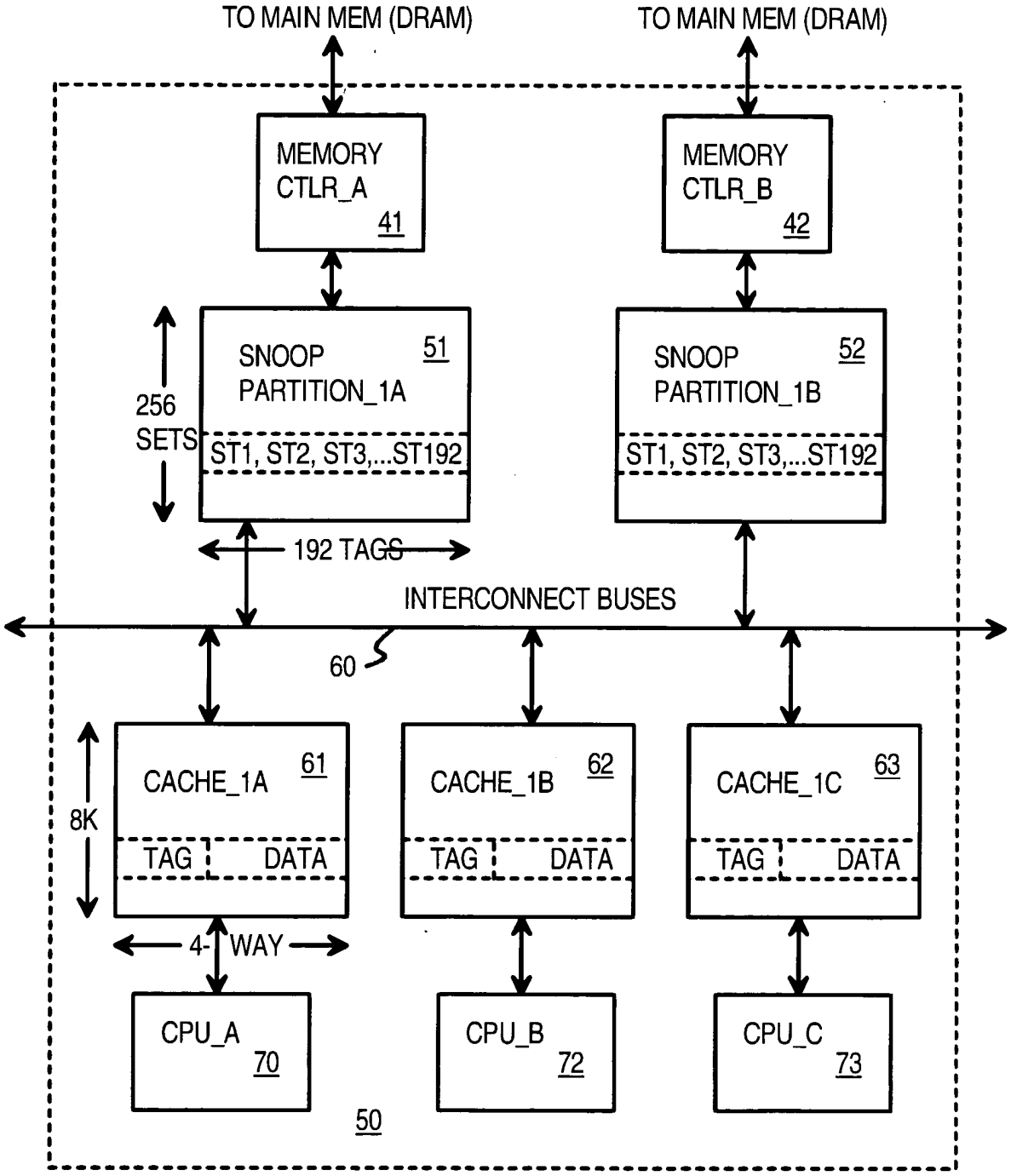

TO MAIN MEM (DRAM)
TO MAIN MEM (DRAM)
MEMORY CTLR_A 41
MEMORY CTLR_B 42
SNOOP PARTITION_1A 51
SNOOP PARTITION_1B 52
256 SETS
ST1, ST2, ST3,...ST192
ST1, ST2, ST3,...ST192
192 TAGS
INTERCONNECT BUSES
60
8K
CACHE_1A 61
CACHE_1B 62
CACHE_1C 63
TAG
DATA
TAG
DATA
TAG
DATA
4-WAY
CPU_A 70
CPU_B 72
CPU_C 73
50

FIG. 4

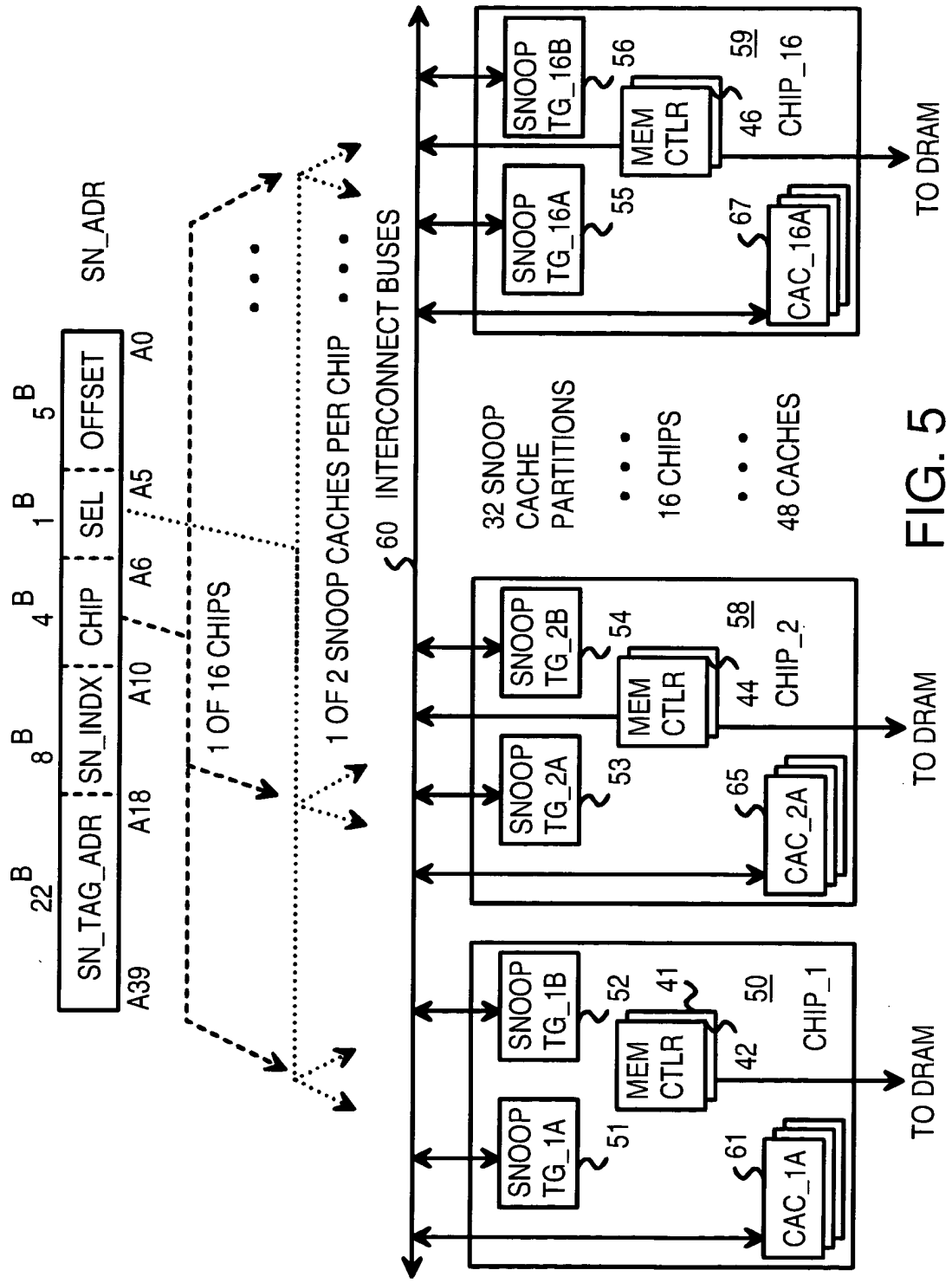
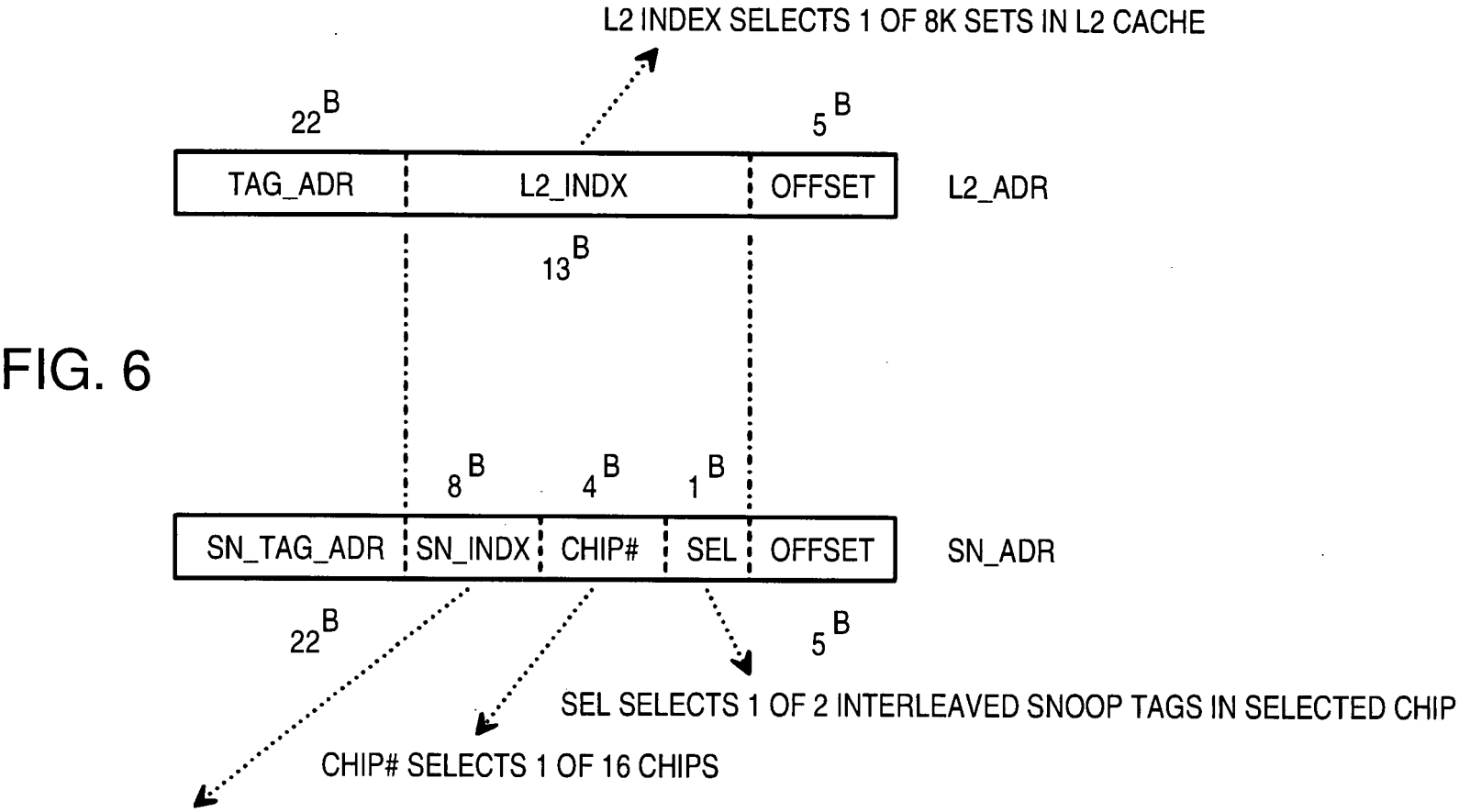

SN_ADR
SN_TAG_ADR
SN_INDX
CHIP
SEL
OFFSET
22 B
8 B
4 B
1 B
5 B
A39
A18
A10
A6
A5
A0
1 OF 16 CHIPS
1 OF 2 SNOOP CACHES PER CHIP
60 INTERCONNECT BUSES
32 SNOOP CACHE PARTITIONS
16 CHIPS
48 CACHES
SNOOP TG_1A
SNOOP TG_1B
MEM CTLR
CHIP_1
50
41
42
51
52
61
CAC_1A
SNOOP TG_2A
SNOOP TG_2B
MEM CTLR
CHIP_2
58
44
53
54
65
CAC_2A
SNOOP TG_16A
SNOOP TG_16B
MEM CTLR
CHIP_16
59
46
55
56
67
CAC_16A
TO DRAM
TO DRAM
TO DRAM

FIG. 5

FIG. 6


L2 INDEX SELECTS 1 OF 8K SETS IN L2 CACHE
22 B
5 B
TAG_ADR
L2_INDX
OFFSET
L2_ADR
13 B
8 B
4 B
1 B
SN_TAG_ADR
SN_INDX
CHIP#
SEL
OFFSET
SN_ADR
22 B
5 B
SEL SELECTS 1 OF 2 INTERLEAVED SNOOP TAGS IN SELECTED CHIP
CHIP# SELECTS 1 OF 16 CHIPS
SN INDEX SELECTS 1 OF 256 SETS IN SELECTED SNOOP TAG PARTITION

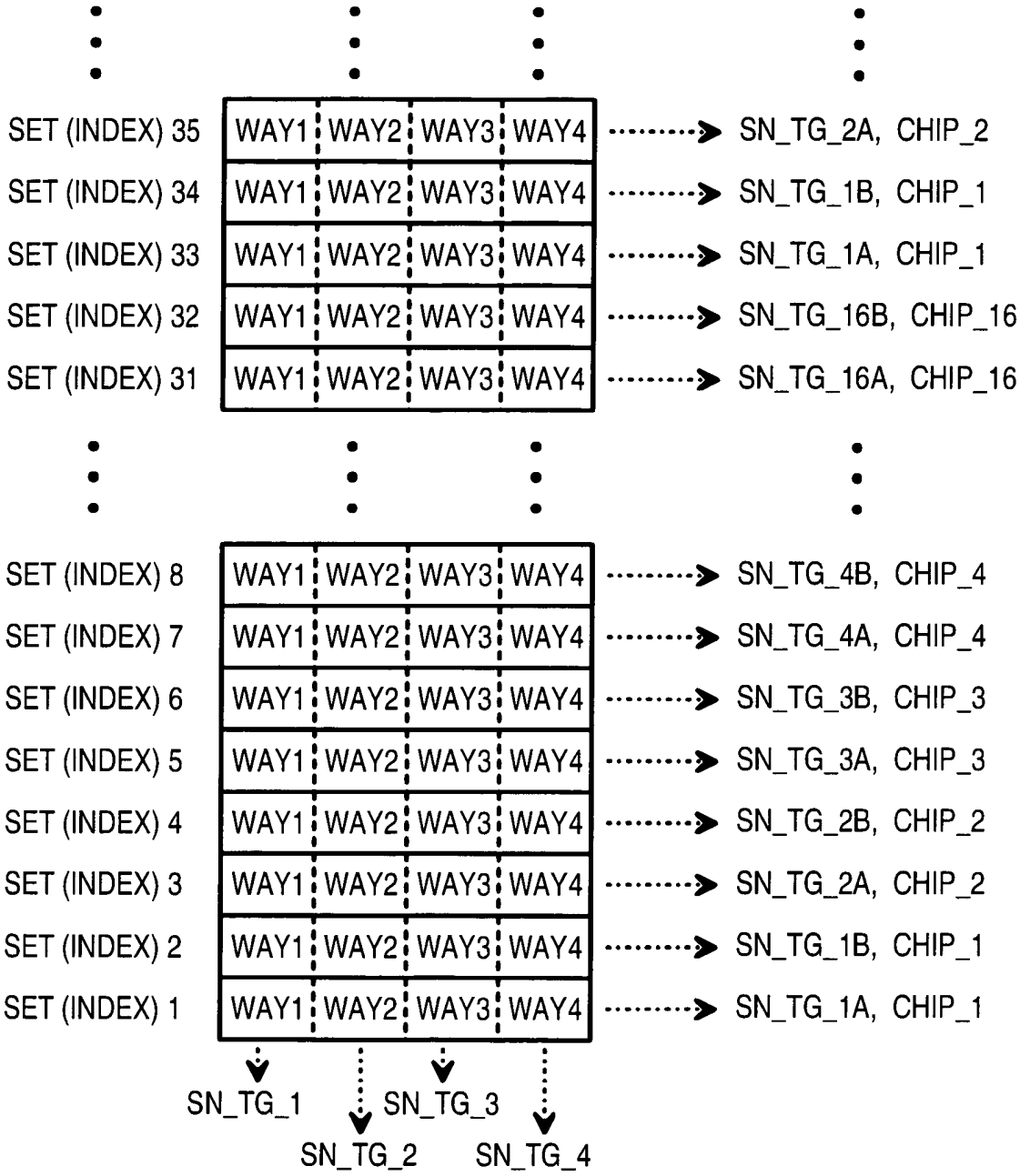

SET (INDEX) 35
SET (INDEX) 34
SET (INDEX) 33
SET (INDEX) 32
SET (INDEX) 31
SET (INDEX) 8
SET (INDEX) 7
SET (INDEX) 6
SET (INDEX) 5
SET (INDEX) 4
SET (INDEX) 3
SET (INDEX) 2
SET (INDEX) 1
WAY1 WAY2 WAY3 WAY4
SN_TG_2A, CHIP_2
SN_TG_1B, CHIP_1
SN_TG_1A, CHIP_1
SN_TG_16B, CHIP_16
SN_TG_16A, CHIP_16
SN_TG_4B, CHIP_4
SN_TG_4A, CHIP_4
SN_TG_3B, CHIP_3
SN_TG_3A, CHIP_3
SN_TG_2B, CHIP_2
SN_TG_2A, CHIP_2
SN_TG_1B, CHIP_1
SN_TG_1A, CHIP_1
SN_TG_1
SN_TG_2
SN_TG_3
SN_TG_4
61
CAC_1A

FIG. 7

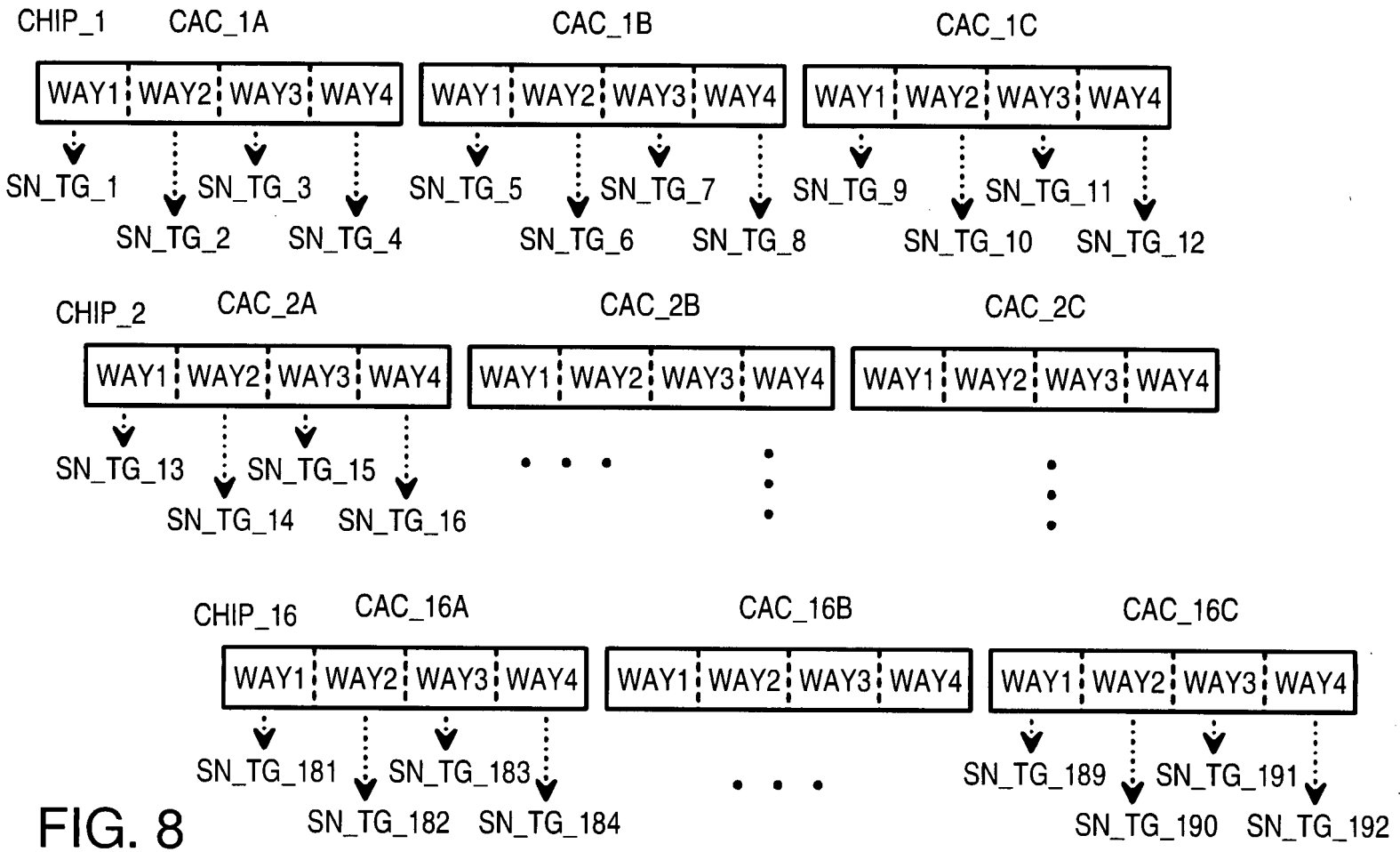


FIG. 8

TOTAL L2 ENTRIES PER SET = 16 CHIPS X 3 L2 CACHES/CHIP X 4 WAY = 192 ENTRIES/SET

TOTAL SN ENTRIES PER SN CACHE INDEX = 192 ENTRIES

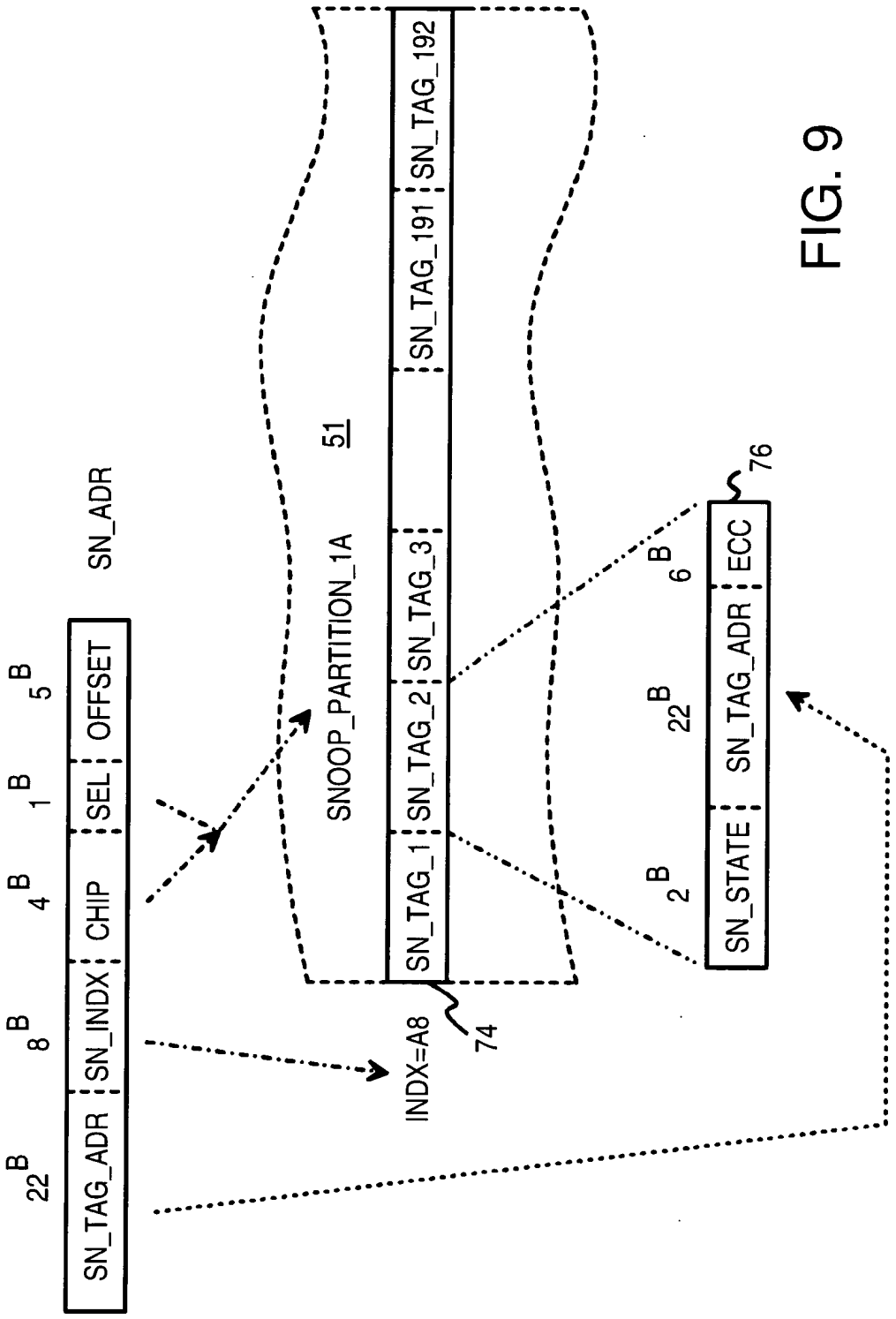

SN_ADR
22B
SN_TAG_ADR
8B
SN_INDX
4B
CHIP
1B
SEL
5B
OFFSET
INDX=A8
74
SNOOP_PARTITION_1A
51
SN_TAG_1
SN_TAG_2
SN_TAG_3
SN_TAG_191
SN_TAG_192
2B
SN_STATE
22B
SN_TAG_ADR
6B
ECC
76

FIG. 9

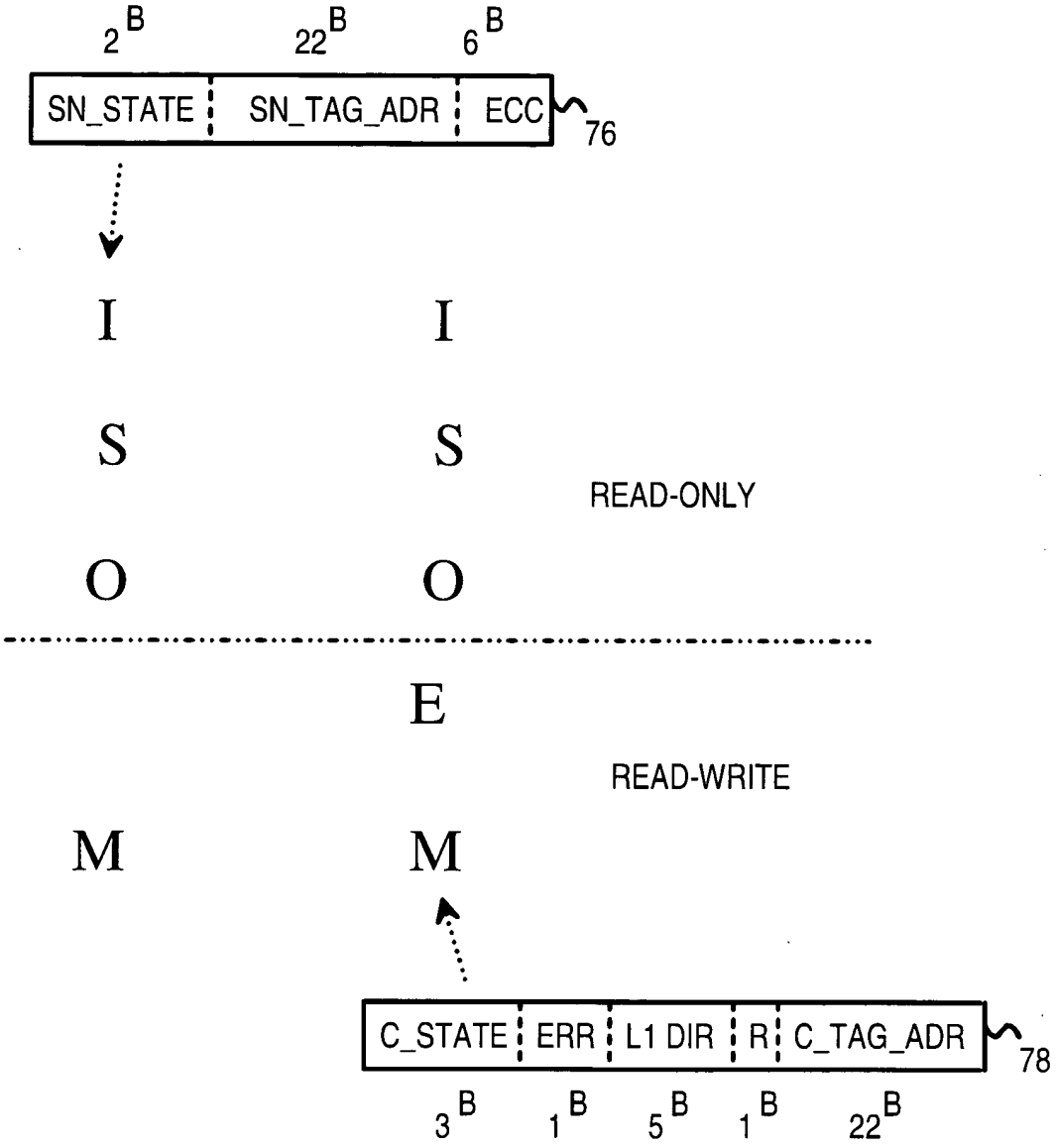

2 B
22 B
6 B
SN_STATE
SN_TAG_ADR
ECC
76
I
I
S
S
READ-ONLY
O
O
E
READ-WRITE
M
M
C_STATE
ERR
L1 DIR
R
C_TAG_ADR
78
3 B
1 B
5 B
1 B
22 B

FIG. 10

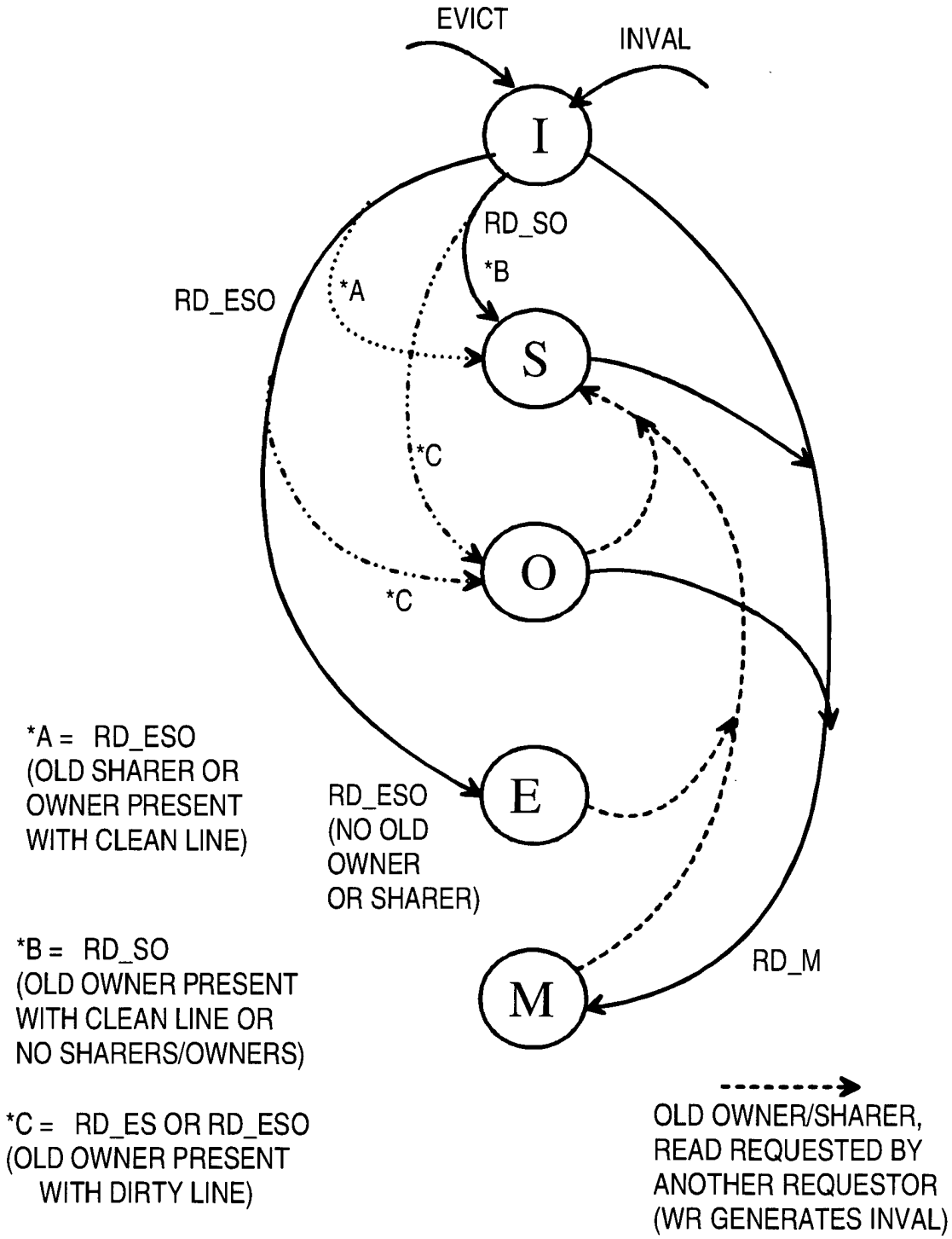

EVICT
INVAL
I
RD_SO
*B
RD_ESO
*A
S
*C
O
*C
E
M
RD_M
*A = RD_ESO
(OLD SHARER OR
OWNER PRESENT
WITH CLEAN LINE)
RD_ESO
(NO OLD
OWNER
OR SHARER)
*B = RD_SO
(OLD OWNER PRESENT
WITH CLEAN LINE OR
NO SHARERS/OWNERS)
*C = RD_ES OR RD_ESO
(OLD OWNER PRESENT
WITH DIRTY LINE)
OLD OWNER/SHARER,
READ REQUESTED BY
ANOTHER REQUESTOR
(WR GENERATES INVAL)

FIG. 11

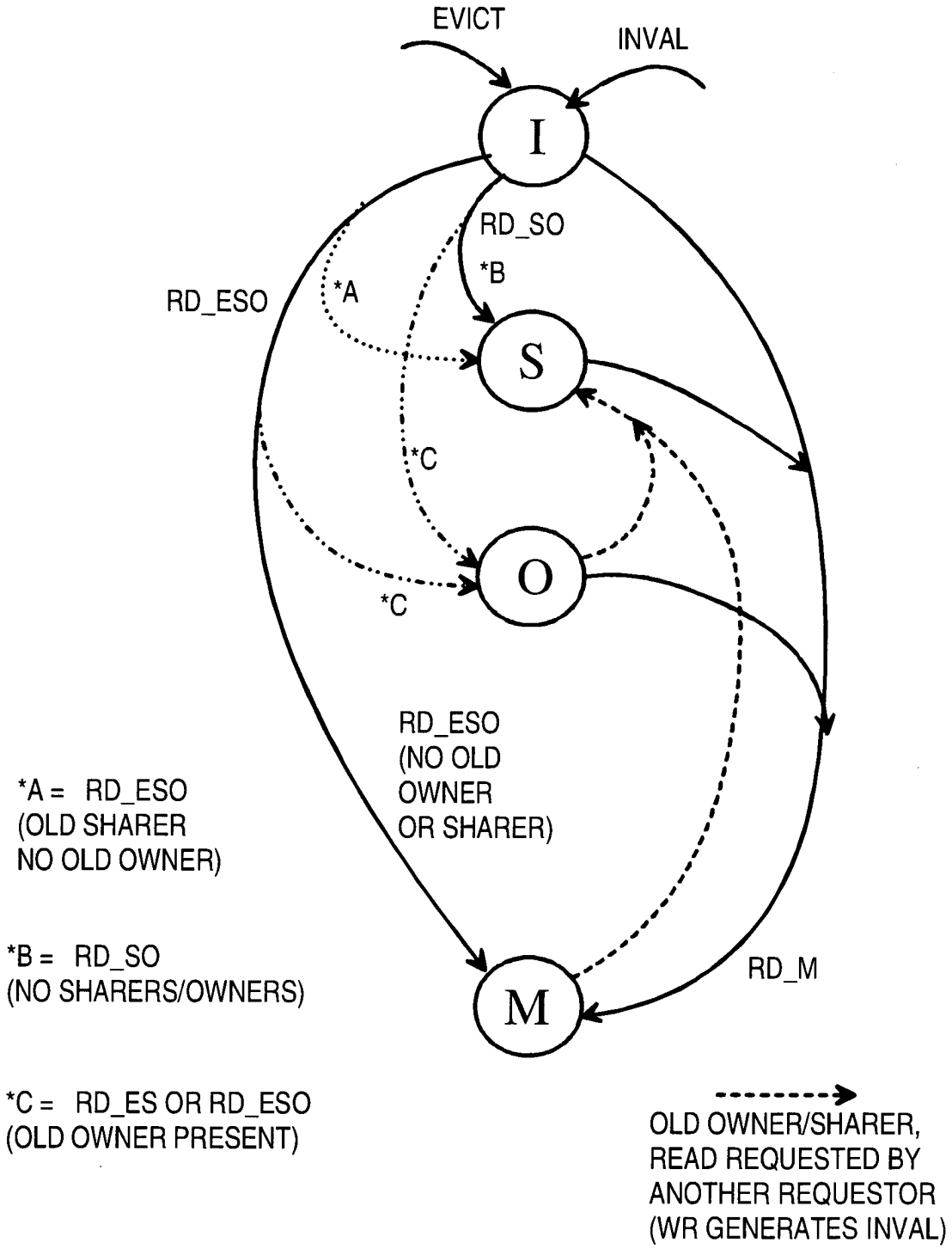

EVICT
INVAL
I
RD_SO
*B
RD_ESO
*A
S
*C
O
*C
RD_ESO
(NO OLD
OWNER
OR SHARER)
*A = RD_ESO
(OLD SHARER
NO OLD OWNER)
*B = RD_SO
(NO SHARERS/OWNERS)
M
RD_M
*C = RD_ES OR RD_ESO
(OLD OWNER PRESENT)
OLD OWNER/SHARER,
READ REQUESTED BY
ANOTHER REQUESTOR
(WR GENERATES INVAL)

FIG. 12

THIS PAGE BLANK (USPTO)